

在第一篇中介绍了利用算法提高数字设计中的性能；第二篇介绍流水线的原理，以及在工业控制中的应用。本章主要介绍如何利用流水线技术提高系统的主频性能。数字设计的流水线技术与工业控制中流水线及其类似，有许多可以借鉴的地方。在下图在算法文章就提到过，利用算法优化设计可以提高系统的性能。但如果在优化之后仍不能解决问题，则会利用流水线技术，继续提供优化。在时序电路设计中，流水线单元可以简单用下图中红框表示部分。即一个组合逻辑单元（如乘法器，加法器，CRC，8b/10 或具备一定功能算法组合电路等）和一个寄存器组。

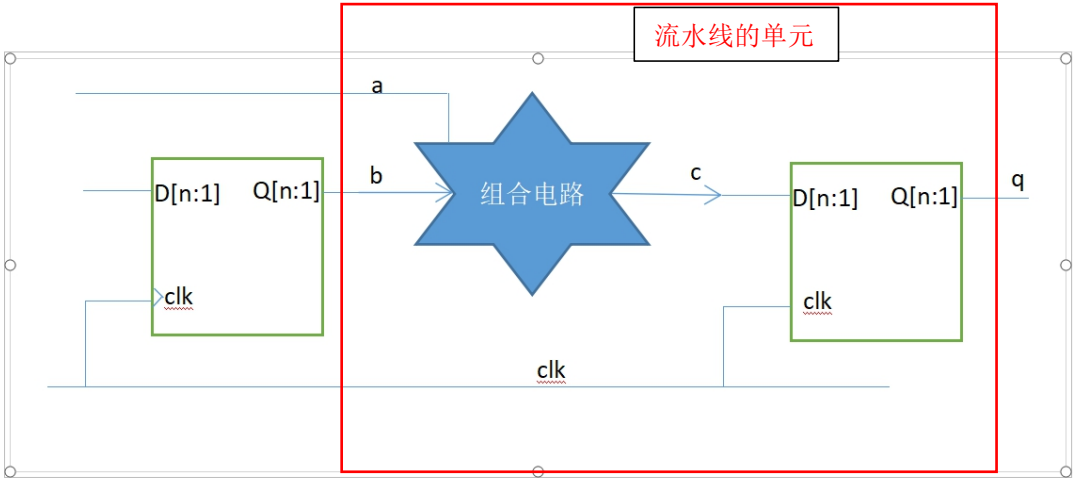


图 1 组合时序电路

在实际的设计优化中也是要分析系统中流水线结构，在复杂流水线中分析出流水线的交叉或合并等结构，找出**关键流水线（影响当前性能的流水线）**，在找出影响该级流水线的**关键部件**（在一条多级流水线中，哪个部件对关键流水线影响最大）。

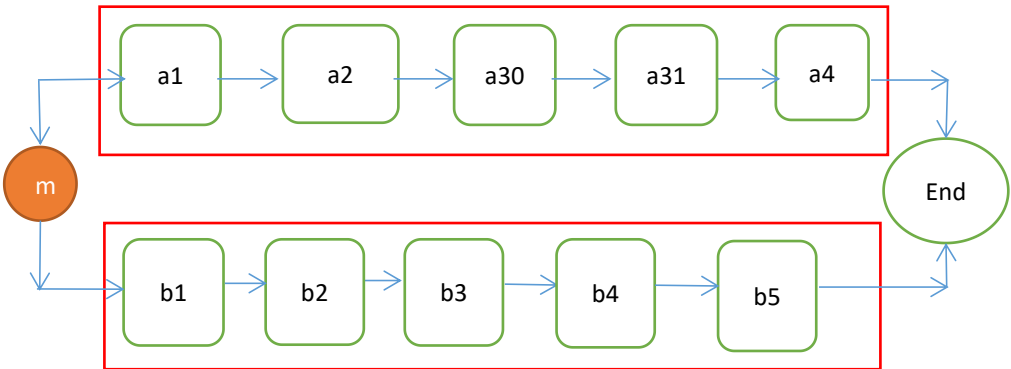
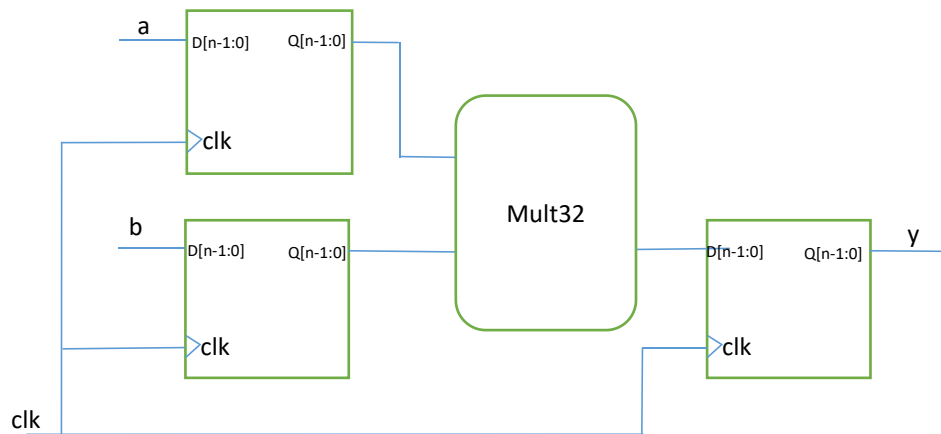


图 2 合并流水线

找到关键路径和关键部件(EDA 工具可以帮助分析)后，就可对关键部件得组合逻辑在逻辑上拆分，插入一条或多条流水线。

下面以 32 位乘法器为例介绍流水线使用以优化逻辑电路设计，电路模型如图 3。为评估方便，我们采用 Verilog 代码进行书写，同时添加 PLL 模组。器件采用 Intel cyclone10LP 在 slow 1200mv 85C Model 得到 84.69MHz。



其 Verilog 代码如下：

```

module mult32
(
    input          clk,
    input          ci,
    input  [31:0]  a,
    input  [31:0]  b,
    output [63:0]  y
);

    wire sys_clk;
    wire locked;
    wire rst=!locked;

    reg  [31:0] a_r,b_r;
    reg  [63:0] y_r;

    always@(posedge sys_clk)
    begin
        a_r<=a;
        b_r<=b;
    end

    always@(posedge sys_clk or posedge rst)
    if(rst) begin
        y_r<=0;
    end
    else
        y_r<=a_r*b_r; //只有一级流水

    assign y=y_r;

```

```

pll1    pll_inst
(
    .areset    (1'b0),
    .inclck0   (clk),      //50Mhz
    .c0        (sys_clk),  //160Mhz
    .locked    (locked)
);
Endmodule

```

我们分析结果，最主要的延迟为 32 位乘法器带来的，由于 32 位乘法器逻辑复杂，多级门电路延迟，因此我们将在 32 位乘法器中插入两级流水线，采用 4 个 16 位乘法器，进行错位相乘、相加的办法。由于原理图的方式绘制比较复杂，因此下面直接采用 Verilog 代码描述的方式进行。

Verilog 代码如下：

```

module mult32
(
    input          clk,
    input          ci,
    input [31:0] a,
    input [31:0] b,
    output [63:0] y

);

    wire sys_clk;
    wire locked;
    wire rst=!locked;

    reg [31:0] a_r,b_r;

    reg [63:0] y_r;

    reg [15:0] a_r0,b_r0; //定义 16 位接口寄存器，同步输入接口
    reg [15:0] a_r1,b_r1; //定义 16 位接口寄存器，同步输入接口

    reg [31:0] y11,y12,y21,y22;//定义 32 位寄存器，作为 16 位乘法的结果

    reg [63:0] ya,yb;

    always@(posedge sys_clk)
    begin
        a_r0<=a[15:0];
        b_r0<=b[15:0];

        a_r1<=a[31:16];
        b_r1<=b[31:16];
    end

```

```

end

always@(posedge sys_clk or posedge rst)
if(rst) begin
    y11 <=0;
    y12 <=0;
    y21 <=0;
    y22 <=0;

end
else
    begin
        //第一级流水
        y11<=a_r0*b_r0;
        y12<=a_r0*b_r1;
        y21<=a_r1*b_r0;
        y22<=a_r1*b_r1;

        //第二级流水
        ya<={31'b0,y11}+{16'b0,y12,16'b0};
        yb<={16'b0,y21,16'b0}+{y22,32'b0};

        //第三级流水，对等原设计的一级
        y_r<=ya+yb;

    end

    assign y=y_r;

pll1    pll_inst
(
    .areset    (1'b0),
    .inclk0    (clk),
    .c0        (sys_clk),
    .locked    (locked)
);

Endmodule

```

最后在同等条件下评估可以得到 173.07MHz. 可见流水线的使用明显提高系统的运行速度。

总结：本文对比工业过程控制中流水的概念，分析在时序逻辑电路中流水线的使用，并以 32 位乘法运算为例，评估了插入流水线后系统性能提升。对比插入流水前后的优缺点如下：

优点：

- (1) 插入流水后，系统的时钟频率明显提升。提高系统性能。
- (2) 在流水线保持连续的情况下，每个时钟周期都可得到一次更新结果。

缺点：

- (1) 时序逻辑变得复杂
- (2) 使用资源明显增多，特别寄存器的数量明显增加

(3) 在流水线连续的情况下，第一个计算结果延迟等于插入的流水级数。

(4) 在流水线断续的情况下，性能可能会降低。而且由于处理不连续的输入，在后期使用结果时，也变得比较复杂，因此是否插入流水应依照系统的要求，以及流水线中的流水断续频率，以统计的结果进行评估。

(5) 在处理多级流水线合并的情况下，其它流水线的流水级数应配齐，增加了系统复杂度。